Amendments to the Specification:

Please replace the table beginning at page 15, line 21 with the following amended table:

	1	2	3	4	5	6	7	8
microstore lookup	n1	cb	n2	XX	b1	b2	b3	b4
reg addr gen		n1	cb	XX	XX	b1	b2	b3
reg file lookup			n1	cb	XX	XX	b1	b2
ALU/shifter/cc				n1	cb	XX	XX	b1
write back			m2		n1	cb	XX	XX

Please replace the table beginning at page 16, line 17 with the following amended table:

	1	2	3	4	5	6	7	8
microstore lookup	n1	cb	n2	XX	b1	b2	b3	b4
reg addr gen		n1	cb	n2	XX	b1	b2	b3
reg file lookup			n1	cb	n2	XX	b1	b2
ALU/shifter/cc				n1	cb	n2	XX	b1
write back					n1	cb	n2	XX

Please replace the table beginning at page 16, line 29 with the following amended table:

	1	2	3	4	5	6	7	8	9
microstore lookup	n1	cb	n2	n3	b1	b2	b3	b4	b5
reg addr gen		n1	cb	n2	n3	b1	b2	b3	b4
reg file lookup			n1	cb	n2	n3	b1	b2	b3
ALU/shftr/cc				n1	cb	n2	n3	b1	b2
write back					n1	cb	n2	n3	b1

Please replace the table beginning at page 17, line 8 with the following amended table:

	1	2	3	4	5	6	7	8
microstore lookup	n1	n2	cb	XX	b1	b2	b3	b4
reg addr gen		n1	n2	cb	XX	b1	b2	b3
reg file lookup			n1	n2	cb	XX	b1	b2
ALU/shifter/cc				n1	n2	cb	XX	b1
write back					n1	n2	cb	XX

Please replace the table beginning at page 17, line 25 with the following amended table:

	1	2	3	4	5	6	7	8
microstore lookup	n1	n2	cb	n3	b1	b2	b3	b4
reg addr gen		n1	n2	cb	n3	b1	b2	b3
reg file lookup			n1	n2	cb	n3	b1	b2
ALU/shifter/cc				n1	n2	cb	n3	b1
write back					n1	n2	cb	n3

Please replace the table beginning at page 18, line 9 with the following amended table:

	1	2	3	4	5	6	7	8
microstore lookup	n1	cb	n1	b1	b2	b3	b4	b5
reg addr gen		n1	cb	XX	b1	b2	b3	b4
reg file lookup			n1	cb	XX	b1	b2	b3
ALU/shifter/cc				n1	cb	XX	b1	b2
write back					n1	cb	XX	b1

Please replace the table beginning at page 18, line 23 with the following amended table:

	1	2	3	4	5	6	7	8
microstore lookup	n1	cb	n1	XX	n2	n3	n4	n5
reg addr gen		n1	cb	n1	XX	n2	n3	n4
reg file lookup			n1	cb	n1	XX	n2	n3
ALU/shifter/cc				n1	cb	n1	XX	n2
write back					n1	cb	n1	XX

Please replace the table beginning at page 19, line 10 with the following amended table:

	1	2	3	4	5	6	7	8
microstore lookup	n1	cb	n1	XX	b1	b2	b3	b4
reg addr gen		n1	cb	XX	XX	b1	b2	b3
reg file lookup			n1	cb	XX	XX	b1	b2
ALU/shifter/cc				n1	cb	XX	XX	b1
write back					n1	cb	XX	XX

Please replace the table beginning at page 19, line 22 with the following amended table:

	1	2	3	4	5	6	7	8
microstore lookup	n1	cb	n2	b1	b2	b3	b4	b5
reg addr gen		n1	cb	n2	b1	b2	b3	b4
reg file lookup			n1	cb	n2	b1	b2	b3
ALU/shifter/cc				n1	cb	n2	b1	b2
write back					n1	cb	n2	b1

Please replace the table beginning at page 20, line 3 with the following amended table:

	1	2	3	4	5	6	7	8	9
microstore lookup	n1	cb	n2	XX	n3	n4	n5	n6	n7
reg addr gen		n1	cb	n2	XX	n3	n4	n5	n6
reg file lkup			n1	cb	n2	XX	n3	n4	n5
ALU/shftr/cc				n1	cb	n2	XX	n3	n4
write back					n1	cb	n2	XX	n3

Please replace the table beginning at page 20, line 20 with the following amended table:

	1	2	3	4	5	6	7	8	9
microstore lookup	n1	cb	n2	XX	b1	b2	b3	b4	b5
reg addr gen		n1	cb	n2	XX	b1	b2	b3	b4
reg file lkup			n1	cb	n2	XX	b1	b2	b3
ALU/shftr/cc				n1	cb	n2	XX	b1	b2
write back					n1	cb	n2	XX	b1

Please replace the table beginning at page 20, line 36 with the following amended table:

	1	2	3	4	5	6	7	8	9
microstore lookup	n1	jp	XX	XX	XX	j1	j2	j3	j4
reg addr gen		n1	jp	XX	XX	XX	j1	j2	j3
reg file lkup			n1	jp	XX	XX	XX	j1	j2
ALU/shftr/cc				n1	jp	XX	XX	XX	j1
write back					n1	jp	XX	XX	XX

Please replace the table beginning at page 21, line 14 with the following amended table:

	1	2	3	4	5	6	7	8	9
microstore lookup	o1	ca	br	n1	n2	n3	n4	n5	n6
reg addr gen		o1	ca	XX	n1	n2	n3	n4	n5
reg file lkup			o1	ca	XX	n1	n2	n3	n4
ALU/shftr/cc				o1	ca	XX	n1	n2	n3
write back					o1	ca	XX	n1	n2

Please replace the paragraph beginning at page 21, line 14 with the following amended paragraph:

Contiguous memory references will be received in queue 90 when the chaining bit is set. Those contiguous references will typically be stored in the order queue 90c because the contiguous memory references are multiple memory references from a single thread. In order to provide synchronization, the memory controller 26a need only signal at the end of the chained memory references when done. However, in an optimized memory chaining, (~~e.g.,~~ e.g., when optimized MEM bit and chaining bit are set) the memory references could go into different banks and potentially complete on one of the banks issuing the signal Adone@ before the other bank was fully drained, thus destroying coherency. Therefore, the chain bit is used by the controller 110 to maintain the memory references from the current queue.

Please replace the paragraph beginning at page 34, line 1 with the following amended paragraph:

The SRAM controller 26b includes a lock lookup device 142 which is an eight (8 entry address content addressable memory for look-ups of read locks. Each position ~~include~~ includes a valid bit that is examined by subsequent read-lock requests. The address and command queue 120 also includes a Read Lock Fail Queue 120d. The Read Lock Fail Queue 120d is used to hold read memory reference requests that fail because of a lock existing on a portion of memory. That is, one of the microengines issues a memory request that has a read lock request that is processed in address and control queue 120. The memory request will operate on either the order queue 120c or the read queue 120b and will recognize it as a read lock request. The controller 26b will access lock lookup device 142 to determine whether this memory location is already locked. If this memory location is locked from any prior read lock request, then this memory lock request will fail and will be stored in the read lock fail queue 120d. If it is unlocked or if 142 shows no lock on that address, then the address of that memory reference will be used by the SRAM interface 140 to perform a traditional SRAM address read/write request to memory 16b. The command controller and address generator 138 will also enter the lock into

the lock look up device 142 so that subsequent read lock requests will find the memory location locked. A memory location is unlocked by operation of [[the]] a microcontrol instruction in a program after the need for the lock has ended. The location is unlocked by clearing the valid bit in the CAM. After an unlock, the read lock fail queue 120d becomes the highest priority queue giving all queued read lock misses, a chance to issue a memory lock request.

Please replace the paragraph beginning at page 3, line 3 with the following amended paragraph:

~~FIG. 2 IS~~ FIGS. 2A and 2B show a detailed block diagram of the hardware-based multithreaded processor of FIG. 1.

FIG. 3C is a block diagram showing general purpose register address arrangement.

~~FIG. 4 is~~ FIGS. 4A and 4B show a block diagram of a memory controller for enhanced bandwidth operation used in the hardware-based multithreaded processor.

FIG. ~~4A~~ 4Cis a flow chart that represents an arbitration policy in an SDRAM controller of FIG. 4.

FIG. ~~4B~~ 4D is a timing diagram that shows advantages of optimizing SDRAM controller.

~~FIG. 5 is~~ FIGS. 5A and 5B show a block diagram of a memory controller for latency limited operations used in the hardware-based multithreaded processor.

FIG. [[5A]] 5C is a timing diagram that shows advantages of optimizing SRAM controller.

~~FIG. 6 is~~ FIGS. 6A, 6B, 6C, and 6D show a block diagram of a communication bus interface in the processor of FIG. 1.

Please replace the paragraph beginning at page 31, line 28 with the following amended paragraph:

Referring to FIG. [[4A]] 4C, a flow representation of the arbitration policy in the SDRAM controller 26a is shown. The arbitration policy favors chained microengine memory requests. The process 115 starts by examining for Chained microengine memory reference requests 115a . The process 115 stays at the chained requests until the chain bit is cleared. The process examines ASB bus requests 115b followed by PCI bus requests 115c, High Priority

Queue Service 115d, Opposite Bank Requests 115e, Order Queue Requests 115f, and Same Bank Requests 115g. Chained request are serviced completely, whereas services 115b-115d are serviced in round robin order. Only when services 115a-115d are fully drained does the process handle services 115e-115g. Chained microengine memory reference requests are when the previous SDRAM memory request has the chain bit set. When the chain bit is set then the arbitration engine simply services the same queue again, until the chain bit is cleared. The ASB is higher priority than PCI due to the severe performance penalty imposed on the Strong arm core when the ASB is in wait state. PCI has higher priority than the microengines due to the latency requirements of PCI. However with other buses, the arbitration priority could be different.

Please replace the paragraph beginning at page 32, line 7 with the following amended paragraph:

As shown in FIG. [[4B]] 4D, typical timing of a memory without active memory optimization and with active memory optimization is shown. As can be seen, the use of active memory optimizations maximizes the use of the bus and thus hides the inherent latency within physical SDRAM devices. In this example, a non-optimized access can take 14 cycles while optimized access can take 7 ~~cyles~~ cycles.

Please replace the paragraph beginning at page 34, line 31 with the following amended paragraph:

As shown in FIG. [[5A]] 5C, typical timing of a static random access memory without active memory optimization and with active memory optimization is shown. As can be seen, grouping reads and writes improves cycletime eliminating dead cycles.